GARDO: Reinforcing Diffusion Models without Reward Hacking

Haoran He [1*] **Yuxiao Ye** [1] **Jie Liu** [2] **Jiajun Liang** [3] **Zhiyong Wang** [4] **Ziyang Yuan** [3]
Xintao Wang [3] **Hangyu Mao** [5] **Meng Wang** [3] **Pengfei Wan** [3] **Ling Pan** [1†]

[1]Hong Kong University of Science and Technology [2]Chinese University of Hong Kong
[3]KlingAI [4]Harbin Institute of Technology, Shenzhen
[5]Institute of Microelectronics, Chinese Academy of Sciences

Abstract

Fine-tuning diffusion models via online reinforcement learning (RL) has shown great potential for enhancing text-to-image alignment. However, since precisely specifying a ground-truth objective for visual tasks remains challenging, the models are often optimized using a proxy reward that only partially captures the true goal. This mismatch often leads to reward hacking, where proxy scores increase while real image quality deteriorates and generation diversity collapses. Prior methods commonly mitigate reward hacking by regularizing the online policy toward a fixed reference model, but uniform KL regularization compromise sample efficiency and impede the exploration of novel, high-reward regions, as the reference is usually sub-optimal. We propose **G**ated and **A**daptive **R**egularization with **D**iversity-aware **O**ptimization (**GARDO**), a versatile framework compatible with various RL algorithms. Departing from standard constrained RL, GARDO reframes the optimization landscape through a unified, dynamic regularization strategy that simultaneously addresses stability and exploration. We propose an uncertainty-gated mechanism that selectively penalizes only high-uncertainty samples to prevent excessive restriction, coupled with an adaptive reference policy that evolves in tandem with the optimized policy to ensure a progressive, relevant regularization target. Furthermore, by integrating a diversity-aware objective that amplifies rewards for distinct, high-quality outputs, GARDO actively addresses mode collapse while maintaining robust training stability. Extensive experiments across diverse proxy rewards and hold-out unseen metrics consistently show that GARDO mitigates reward hacking and enhances generation diversity without sacrificing sample efficiency or exploration, highlighting its effectiveness and robustness.

1 Introduction

Denoising diffusion and flow models trained on large-scale datasets have found great success in text-to-image generation tasks, exhibiting unprecedented capabilities in visual quality [46, 26, 66, 69]. While supervised pre-training provides these models a strong prior, it is often insufficient for ensuring alignment with human preferences [40, 56]. Reinforcement Learning (RL) has emerged as a predominant paradigm to address this gap, either employing online policy-gradient approaches [3, 10, 34, 15, 38, 72] or direct reward backpropagation [5, 70]. These methods typically assume a reward model $r(x)$ that captures human preferences, and the training objective is to maximize the rewards over generated samples, i.e., $\mathbb{E}_{x\sim\pi}[r(x)]$. Consequently, the accuracy of $r(x)$ plays a critical role in fine-tuning performance. However, a key challenge arises from the nature of reward specification in vision. In contrast to the language domain, where rewards are often verifiable (e.g., the correctness

*This work was conducted during the author's internship at KlingAI.
†Correspondence to: Ling Pan (lingpan@ust.hk).

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

of mathematical solutions or compiled code), rewards for visual tasks are typically more complex. Such rewards typically fall into two categories, both of which are imperfect proxies for genuine human preference: 1) *Model-based rewards* such as ImageReward [70], UnifiedReward [62], and HPSv3 [37] are trained on finite human preference datasets to approximate a ground-truth "genuine" reward, indicating that they are accurate only within their training data distribution; 2) *Rule-based rewards* such as object detection and text rendering (i.e., OCR) [4] are limited to evaluating specific attributes, and thus fail to capture the overall qualities of generated samples. These limitations expose a significant distribution shift [64] vulnerability: fine-tuning methods can rapidly over-optimize the proxy reward by generating images that fall outside the trusted support of the reward model. This often yields spurious reward signals and thereby leads to *reward hacking* [54], where proxy rewards increase while the actual quality of the images degrades.

To address this challenge, prior work has employed KL-based regularization during fine-tuning to mitigate over-optimization on these spurious reward signals [57, 10, 34]. However, as the online policy improves, the divergence from the static reference model can cause the KL penalty to dominate the RL loss. This often leads to diminishing policy updates, thereby impeding sample efficiency. Moreover, by design, the regularization constrains the online policy to remain in proximity to the reference model, which is often suboptimal. This can stifle effective exploration and prevent the discovery of emerging behaviors that are absent from the reference model.

Can we prevent reward hacking without compromising sample efficiency and effective exploration? In this paper, we propose **G**ated and **A**daptive **R**egularizat-ion with **D**iversity-aware **O**ptimization (**GARDO**), a novel framework designed to overcome these limitations. Our gated regularization mechanism is motivated by the core insight: **KL penalty is not universally required** – theoretically, only samples assigned with spurious rewards need regularization to prevent hacking. GARDO addresses reward hacking along three complementary axes: *when* to regularize, *where* to regularize, and *how* to preserve diversity. Accordingly, our gated mechanism applies the KL penalty selectively, targeting only a small subset (e.g., $\approx$10%) of samples within each batch that exhibit the highest reward uncertainty [2]. We quantify this uncertainty by measuring the disagreement among an ensemble of reward functions, which serves as an effective proxy for the trustworthiness of the reward signal. High disagreement signifies that a generated sample likely falls into an out-of-distribution region where the reward models are extrapolating unreliably. To further accelerate training and sustain exploration, GARDO introduces a moving reference anchor, periodically refreshing the reference model with a recent snapshot of the online policy, facilitating continual improvement while retaining the stabilizing benefits of KL regularization. Moreover, to enhance mode coverage, we introduce a diversity-aware optimization strategy. This is achieved by amplifying the advantage for high-quality samples that also exhibit high diversity. This amplification is carefully calibrated to ensure it neither dominates nor reverses the sign of the original advantage, which encourages broader exploration without destabilizing the optimization.

Our contributions are summarized as follows: (i) We provide a systematic analysis of the reward hacking phenomenon in RL-based fine-tuning for text-to-image alignment, identifying the core limitations of existing regularization techniques. (ii) We make a novel innovation on traditional KL regularization. First, a gated regularization method selectively applies penalties only to high-uncertainty samples, thereby allowing the majority of samples to be optimized freely towards high-reward regions. Second, an adaptive regularization target facilitates continual improvement and sustained exploration by dynamically updating the reference anchor. (iii) We propose a robust and generalist diversity-aware approach to improve generation diversity and mode coverage for diffusion RL. (iv) Extensive experiments on multiple tasks across different proxies and unseen metrics demonstrate both the efficiency and effectiveness of our proposed method. Our work demonstrates that it is possible to effectively balance the competing demands of sample efficiency, exploration, and diversity, while robustly mitigating reward hacking during the fine-tuning of image generative models. Code is available at `https://anonymous.4open.science/r/GARDO-6BED`.

2 Related Works

Fine-Tuning Diffusion Models via Rewards. Recent research has increasingly focused on fine-tuning pre-trained diffusion models using reward signals derived from human feedback. A reward model (RM) is trained on a finite dataset, learning to assign a scalar score that approximates true human preference [70, 35, 67, 63]. Typical methods include policy-gradient RL [3, 11, 38], direct preference optimization (DPO) [58, 35, 73, 32, 76, 77, 36], and direct reward backpropagation [43, 5,

70]. Recently, Flow-GRPO [34] and DanceGRPO [72] have adapted GRPO for finetuning cutting-edge flow matching models, showing strong performance and inspiring subsequent research [29, 18, 61, 59, 12, 75, 80, 71, 55, 7, 28, 78, 48, 44, 22, 31, 42, 21].

Reward Hacking. Fine-tuning with either model-based or rule-based reward models for alignment of visual generation is highly susceptible to reward hacking [27, 54]. This issue arises because the reward model is an imperfect proxy for the true human preference. As a generative model optimizes against this proxy, it often discovers adversarial solutions that exploit the reward model's flaws, such as favoring extreme saturation or visual noise [5, 35], to achieve a high score, despite failing to satisfy human intent. To address this, standard methods like DPOK [10] and Flow-GRPO [34] employ KL regularization to constrain the policy update. While this can prevent the most egregious forms of reward hacking, it often sacrifices convergence speed and hinders effective exploration. Another line of work, exemplified by RewardDance [67], focuses on improving the reward model itself through scaling; however, this does not eliminate the vulnerability to out-of-distribution samples. Pref-GRPO [61] utilizes a pairwise preference model to provide a more robust reward signal and avoid illusory advantages. This approach, however, is not universally applicable as it is limited to preference-based reward models and incurs substantial computational costs for pairwise comparisons. While recent works [60, 33] introduce auxiliary techniques tailored to GRPO for mitigating reward hacking, they still excessively exploit hackable reward landscape during prolonged training.

3 Preliminaries

3.1 Denoising as an MDP

Both diffusion models and flow models map the source distribution, often a standard Gaussian distribution, to a true data distribution p_0. As shown in previous works [23, 53, 17], these models can utilize an SDE-based sampler during inference to restore from diffused data.

Inspired by DDPO [3], we formulate the multi-step denoising process in flow and diffusion-based models as a Markov Decision Process (MDP), defined by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \rho_0)$. At each denoising step t, the model receives a state $s_t \triangleq (c, t, x_t)$, and predicts the action $a_t \triangleq x_{t-1}$ based on the policy $\pi(a_t|s_t) \triangleq p_\theta(x_{t-1}|x_t, c)$. A non-zero reward $R(s_t, a_t) \triangleq r(x_0, c)$ is given only at the final step, where $R = 0$ if $t \neq 0$. The transition function $\mathcal{P}$ is deterministic. At the beginning of each episode, the initial state s_T is sampled from the initial state distribution ρ_0. The goal is to learn a policy $\pi^* = \arg\max_\pi \mathbb{E}_{\rho_0, a_t \sim \pi(s_t)}\big[R(s_0, a_0)\big]$ by maximizing the expected cumulative reward R.

3.2 RL on Diffusion and Flow Models

Reinforcement Learning (RL) [3, 10] has been widely used for enhancing sample quality by steering outputs towards a desired reward function. Unlike earlier approaches based on PPO [50] or REINFORCE [65], recent methods [34, 72] have demonstrated greater success using GRPO [51]. We take GRPO as the base RL algorithm throughout our paper. GRPO rollouts $\{x_0^i\}_{i=1}^G$ samples conditioned on the same input c, and estimates the advantage within each group:

$$A_t^i = \frac{R(x_0^i) - \text{mean}(\{R(x_0^i)\}_{i=1}^G)}{\text{std}(\{R(x_0^i)\}_{i=1}^G)}, \forall t. \tag{1}$$

Then the following surrogate objective optimizes π_θ:

$$J(\theta) = \mathbb{E}_{x_0 \sim \pi_{\theta_{\text{old}}}}\Big[\frac{1}{G}\sum_{i=1}^G \frac{1}{T}\sum_{t=0}^T \Big(\min\big(\mathrm{r}_t^i A_t^i, \text{clip}(\mathrm{r}_t^i, 1-\epsilon, 1+\epsilon)A_t^i\big) - \beta D_{KL}(\pi_\theta|\pi_{\text{ref}})\Big)\Big], \tag{2}$$

with $\mathrm{r}_t^i = \pi_\theta(a_t|s_t)/\pi_{\theta_{\text{old}}}(a_t|s_t)$ the importance sampling ratio, $\pi_{\theta_{\text{old}}}$ the behavior policy to sample data, ϵ the clipping range of r_t, and D_{KL} the KL regularization term.

4 Method

We first present the reward hacking issues in image generation as a motivational example. Subsequently, we provide a theoretical analysis, and propose practical methods to address the problem. Finally, we provide a didactic example for empirical validation to illustrate our method clearly.

4.1 Reward Hacking in RL-Based Fine-tuning

We investigate RL in the context of fine-tuning text-to-image models. Here, the reward function is often a neural network trained on finite preference data or a rule-based metric, which serves as just a

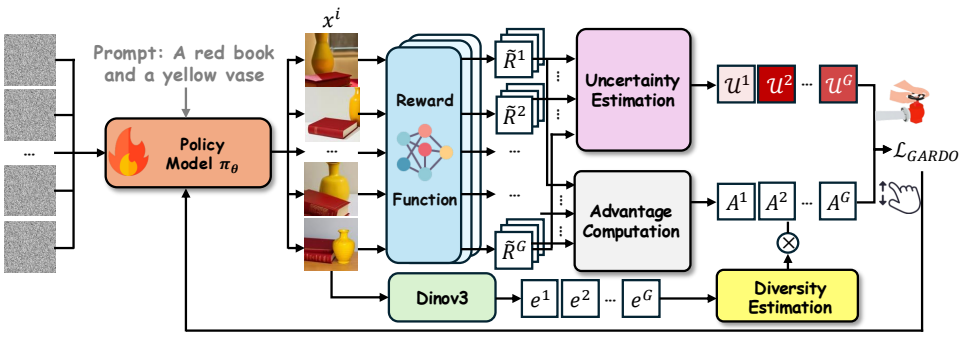


Figure 1: Overview of GARDO. GARDO introduces an uncertainty-driven, gated KL mechanism to control the proportion of regularization, avoiding unnecessary penalties. Our proposed diversity-aware advantage shaping effectively encourages exploration of novel states.

proxy for the true reward. Misalignment between the two objectives can lead to *reward hacking*: a learned policy performs well according to the proxy reward but not according to the true reward [41].

Definition 1 (Reward hacking [27]). *Suppose $\tilde{R}$ is a proxy reward used during RL fine-tuning, and R denotes the true reward. $\tilde{R}$ is a hackable proxy with respect to π_{ref} s.t.*

$$J(\pi, R) < J(\pi_{\text{ref}}, R) \text{ for some } \pi \in \arg\max_{\pi} J(\pi, \tilde{R}), \tag{3}$$

where $J(\pi, R) = \mathbb{E}_\pi[\sum_t R(s_t, a_t)]$ is the RL objective, and π_{ref} is the reference policy.

As demonstrated in Figure 2, standard methods like Flow-GRPO [34] maximize the proxy score (i.e., OCR) at the expense of perceptual quality, failing to align with the true human preferences. Previous methods [57] address this issue by incorporating a KL divergence regularization term,

$$J_\beta(\pi_\theta) = \mathbb{E}_{\pi_\theta}\big[\sum_t \tilde{R}(s_t, a_t)\big] - \beta D_{\text{KL}}(\pi_\theta | \pi_{\text{ref}}), \tag{4}$$

where β controls the regularization strength. It mitigates reward hacking (or reward over-optimization) by explicitly penalizing significant deviations of the online policy, π, from the reference policy, π_{ref}. However, this strategy introduces two significant limitations. First, because the reference policy π_{ref} is typically suboptimal, the regularization loss can impede learning, leading to poor sample efficiency. Second, the constant penalty constrains π_θ to exploit the "safe" region around π_{ref}, preventing it from discovering novel behaviors or solutions that are absent from the reference model.

Ideally, for fine-tuning image generative models via RL, our goal is to reconcile the following competing objectives: (i) optimizing π_θ with high sample efficiency, (ii) avoiding reward hacking and over-optimization, (iii) encouraging exploration to high-reward, novel modes which may be missing from π_{ref}, and (iv) preserving generation diversity. Conventional KL regularization is effective in accomplishing (ii) and (iv), but often at the expense of (i) and (iii). This inherent trade-off motivates our work. Based on our theoretical findings, we propose **G**ated and **A**daptive **R**egularization with **D**iversity-aware **O**ptimization (**GARDO**), a novel framework that is simple, effective, and capable of satisfying all these objectives simultaneously. Overview of GARDO is illustrated in Fig. 1, and we detail our method in the subsequent sections.



Figure 2: Setting OCR as the proxy reward, vanilla RL methods like Flow-GRPO exploit the OCR reward at the cost of losing real image quality, leading to reward hacking. It generates unrealistic, noisy images with blurry backgrounds and visual artifacts. In contrast, our method maintains better image quality and diversity. Prompt:"A storefront with 'GARDO' written on it".

4.2 Gated and Adaptive Regularization

First, we investigate the reasons of reward hacking. As discussed in previous works [16, 68, 56], the optimal solution to the KL-regularized objective in Eq (4) can be written as

$$p^*(x) = \frac{1}{\mathcal{Z}}\pi_{\text{ref}}(x)\exp\left(\frac{\tilde{R}(x)}{\beta}\right), \tag{5}$$

where $\mathcal{Z}$ is an intractable normalization constant [45]. Here, $p^*(x)$ is largely defined by both the reference model $\pi_{\text{ref}}(x)$ and the proxy reward $\tilde{R}(x)$.

Proposition 1 *The probability ratio between any two samples, x^1 and x^2, under the optimal solution distribution defined in Eq. (5) is given by the following closed form,*

$$\frac{p^*(x^1)}{p^*(x^2)} = \frac{\pi_{\text{ref}}(x^1)}{\pi_{\text{ref}}(x^2)}\exp\left(\frac{\tilde{R}(x^1) - \tilde{R}(x^2)}{\beta}\right). \tag{6}$$

If $\pi_{\text{ref}}(x^1) = \pi_{\text{ref}}(x^2)$, then their probability difference is defined solely by the proxy reward, i.e., $p^*(x^1)/p^*(x^2) = \exp((\tilde{R}(x^1) - \tilde{R}(x^2))/\beta)$. Reward hacking arises when the proxy reward is misaligned with the true reward, R. For example, if $\tilde{R}(x^1) > \tilde{R}(x^2)$ while $R(x^1) < R(x^2)$, the model will incorrectly assign a higher sampling probability to the lower-quality sample, x^1. However, in other cases, such as '$x^1 > x^2$' holds for both $\tilde{R}$ and R, the proxy is reliable and does not lead to reward hacking. Inspired by this finding, we have the following key insight:

Takeaway 1

Regularization is not universally required. It is required only for samples with spurious proxy reward $\tilde{R}$.

KL regularization should be conditional on the alignment between proxy and true rewards. Universally penalizing all samples introduces unnecessary regularization signals, thereby impeding convergence.

Gated KL mechanism. KL is gated for selective samples. We propose an uncertainty-driven approach to select samples for regularization during the training process [1]. Only samples that exhibit high uncertainty $\mathcal{U}$ will be penalized, where $\mathcal{U}$ reflects the trustworthiness of the proxy reward [6]. Our approach for quantifying uncertainty is inspired by prior work in reinforcement learning [2], where the uncertainty is often estimated using the disagreement within an ensemble learned K bootstrapped Q-functions Q^k [39], i.e., $\mathcal{U}(s_t, a_t) := \text{Std}(Q^k(s_t, a_t))$. In the denoising MDP of image generation, considering the proxy $\tilde{R}$ is a function of the final state x_0, we can simplify the uncertainty quantification as $\mathcal{U}(s_t) := \text{Std}(V^k(s_t))$. However, directly learning ensemble K value functions from scratch is computationally prohibitive for large-scale generative models, given their vast state space and the complexity of the image generation process. To circumvent this challenge, we adopt a more practical approach that approximates the value function using readily available, pre-trained reward models [30]. Instead of leveraging the deviation among ensemble value functions for estimating uncertainty, we propose a new uncertainty quantification approach:

$$\mathcal{U}(x^1) := w(\tilde{R}(x^1)) - \text{mean}(\{w(\hat{R}_n(x^1))\}_{n=1}^K), \tag{7}$$

where $w(y^i) = \frac{1}{B}\sum_{j\neq i}\mathbb{I}(y^i > y^j)$ denotes the win rate within a batch of size B, and $\{\hat{R}_n\}_{n=1}^K$ are the auxiliary reward models. Under this formulation, a high uncertainty score arises when $w(\tilde{R}) \gg \text{mean}(\{w(\hat{R}_n)\}_{n=1}^K)$, effectively flagging samples with anomalously high proxy $\tilde{R}$ compared to the ensemble. We choose light-weight Aesthetic [49] and ImageReward [70] as $\hat{R}$ throughout our paper (i.e., K=2), thereby the computation cost is negligible. Note that only $\tilde{R}$ is optimized during fine-tuning, while $\hat{R}$ only serves as a metric for estimating uncertainty. Surprisingly, a key empirical finding of our work is that this selective penalization is highly efficient. We find that applying the KL penalty to only a small subset of samples (e.g., approximately 10 %) with the highest uncertainty is sufficient to prevent reward hacking.

Moving Reference Anchor. While the gated KL mechanism penalizes only high-uncertainty samples (which answers the *when* to regularize question), we observe that the sample efficiency still slows

as training progresses. As indicated in Proposition 1, π_{ref} also plays a pivotal role in determining the optimal distribution. If $\tilde{R}(x^1) = \tilde{R}(x^2)$, a failure mode arises if $\pi_{\text{ref}}(x^1) \ll \pi_{\text{ref}}(x^2)$, resulting $p^*(x^1) \ll p^*(x^2)$, which is not an expected behavior (i.e., $p^*(x^1) = p^*(x^2)$) [16]. We remark that a static reference model becomes increasingly sub-optimal, particularly at later training stages, which can affect the optimization a lot. To mitigate this limitation and facilitate sustained improvement, we introduce a moving reference anchor of the regularization objective that addresses the *where* to regularize question. We periodically hard-reset the reference model π_{ref} to the current policy at specific epochs, allowing it to remain updated. In particular, the reference model is updated whenever the KL divergence D_{KL} surpasses a pre-defined threshold ϵ_{KL}, or, failing that, after a maximum of m gradient steps. Our proposed moving reference anchor ensures the regularization target remains relevant, preventing the KL penalty from dominating the RL loss and halting exploration, thereby achieving sustained policy improvement. Implementation details are provided in Appendix A.

Takeaway 2

A static reference model inevitably becomes a constraint on RL optimization. Dynamically updating the reference model facilitates prolonged improvement.

4.3 Diversity-Aware Optimization

As noted by Flow-GRPO [34], a significant consequence of reward hacking is the reduced diversity. This issue is exacerbated by the intrinsically mode-seeking nature of reinforcement learning, which often struggles to capture multimodal distributions [24]. Enhancing sample diversity is therefore critical not only for preventing mode collapse but also for broadening the policy's exploration space. While our gated and adaptive regularization scheme effectively mitigates reward hacking without sacrificing sample efficiency, it does not explicitly promote generation diversity. To address this limitation, we introduce a diversity-aware optimization strategy that amplifies rewards for high-quality samples that also exhibit high diversity.

Diversity-aware Advantage Shaping. The core idea is to reshape the advantage function by incorporating a diversity-based signal during policy optimization. Specifically, for a group of generated samples, $\{x_0^i\}_{i=1}^G \sim p_\theta(x_0|c)$, we first map each sample from the pixel space into a semantic feature space, obtaining feature embeddings e^i for each clean image x_0^i. We employ DINOv3 [52] for the feature extraction, which serves as a powerful vision foundation model. A sample's diversity is then quantified by its isolation in the feature space. We define a diversity score, d_i, as the cosine distance to its nearest neighbor within the group, $\{e^i\}_{i=1}^G$. This diversity score is subsequently used to reshape the advantages, as illustrated as follows:

$$e_i = \text{Dinov3}(x_0^i),\ d_i = \min(\{1 - \tfrac{e_i \cdot e_j}{|e_i||e_j|}\}_{j \neq i}),\ \ A_i^{\text{shaped}} = A_i \cdot \text{d}_i \ \text{ if } A_i > 0 \text{ else } A_i,\ \ i \sim [1, G]. \quad (8)$$

There are several key design principles for our proposed diversity-aware advantage reshaping: (1) We use a multiplicative re-weighting of the advantage term rather than an additive diversity bonus. This design circumvents the need for delicate hyperparameter tuning to balance the scales of the proxy reward and the diversity score, which could otherwise cause one signal to dominate the other. (2) The advantage shaping is applied only when a sample's advantage is positive. This is a critical constraint that ensures the model is rewarded only for generating samples that are both high-quality and novel. It explicitly prevents the model from generating aberrant or low-quality images simply to increase its diversity score. Our method is summarized in Alg. 1 in Appendix A.

Takeaway 3

Multiplicative advantage reshaping exclusively within positive samples enables robust diversity improvement.

Empirical Validation. As illustrated in Fig. 3, we provide a didactic example to validate the superior efficacy of our method. We observe that only GARDO successfully captures all high-reward modes, reaching the maximum reward. The most notable outcome is GARDO's discovery of the central cluster, a mode with only $0.1\times$ probability density assigned by the reference model compared with other modes. This "mode recovery" capacity underscores GARDO's potential for robust exploration, showing it can incentivize emerging behaviors that lie far from the pre-trained distribution.

An Interesting Finding. Beyond the above techniques, we find that simply removing the standard deviation in advantage normalization also helps mitigate reward hacking. In image generation tasks,

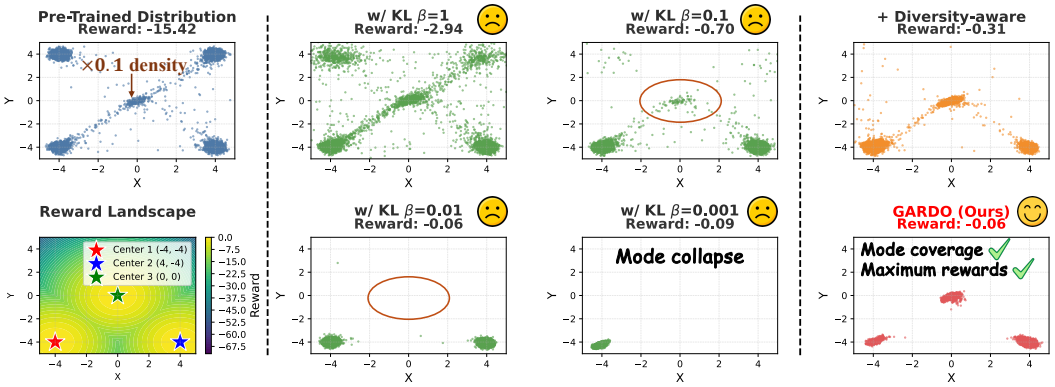


Figure 3: We train a diffusion model with 3-layer MLP on Gaussian mixtures (pre-trained distribution), with the goal to capture multimodal high-reward clusters as shown in the reward landscape. The vanilla RL method (DDPO [3]) with a large KL coefficient β is overly constrained and fails to increase rewards. Conversely, a small β incurs severe mode collapse. Our proposed diversity-aware optimization, when applied alone, successfully captures the multimodal modes, including the central cluster with the lowest probability density in the reference policy π_{ref}. Our full GARDO framework simultaneously achieves maximum reward and discovers all high-reward clusters.

reward models often assign overly similar rewards $R(x_0^i, c)$ to comparable images within the same group, causing an extremely small standard deviation, i.e., $\text{Std} \to 0$. This dangerously amplifies small, and often meaningless, reward differences in Eq. (1), making training sensitive to reward noise and leading to over-optimization. While Pref-GRPO [61] proposes to mitigate this issue by using a preference model to convert rewards into pairwise win-rates, this method is computationally expensive and lacks generality, as it relies on exhaustive pairwise comparisons and is restricted to preference-based reward models. We provide a comparison with Pref-GRPO in Table 5.

5 Experiments

Setup. Following Flow-GRPO [34], we choose SD3.5-Medium [9] as the base model and GRPO as the base RL algorithm for empirical validation. To demonstrate the versatility of our framework across different base models and RL algorithms, we also provide supplementary results on Flux.1-dev [26] and on a distinctly different online RL algorithm, DiffusionNFT [79], in Appendix B.2. Throughout all experiments, images are generated at 512×512 resolution. We fine-tune the reference model with LoRA [20]. We set the group size $G = 24$ for estimating the diversity-aware advantages.

Benchmarks. We employ multiple tasks with diverse metrics to evaluate the performance of our method for preventing reward hacking without sacrificing sample efficiency. We employ GenEval [13] and Text Render (OCR) [4] as the proxy tasks. GenEval evaluates the model's generation ability on complex compositional prompts, including 6 different dimensions like object counting, spatial relations, and attribute binding. OCR measures the text accuracy of generated images. For these two tasks, we use the corresponding training and test sets from Flow-GRPO. We employ unseen metrics including Aesthetic [49], PickScore [25], ImageReward [70], ClipScore [19], and HPSv3 [37] for assessing the o.o.d. generalization performance. For evaluating the diversity of the generated images, we use the mean of pairwise cosine distance across a group of images for quantification, i.e., $\text{Div} = \text{mean}_{i,j\in[1,G],i\neq j}(1 - \frac{e_i \cdot e_j}{|e_i||e_j|})$. More implementation details are provided in Appendix A.

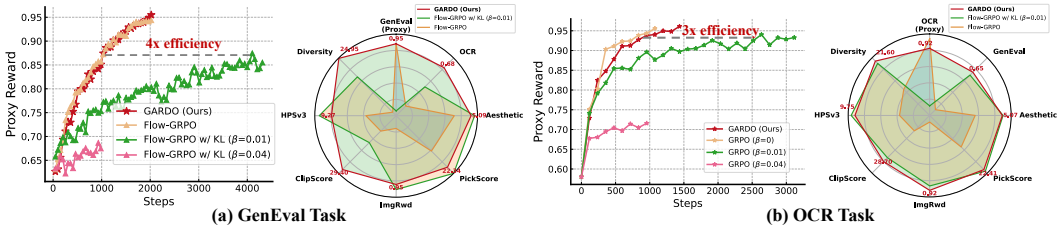
5.1 Results Analysis

Removing Std from Advantage Normalization is Useful. As shown in Table 1 (GenEval) and Table 3 (OCR), eliminating standard deviation normalization alleviates reward hacking and improves performance on unseen rewards compared to the baseline, while largely preserving sample efficiency and high proxy rewards. However, its performance on unseen metrics still falls short of the reference model. This observation clearly demonstrates that, while this technique is valuable, it is insufficient on its own to fully resolve the reward hacking problem as defined in Definition 1.

Efficiency vs. Reward Hacking. The results in Table 1 (GenEval) and Table 3 (OCR) highlight a critical trade-off inherent in RL fine-tuning. The GRPO ($\beta = 0$) baseline, while achieving a high proxy reward, suffers from severe reward hacking, as evidenced by its poor performance on unseen metrics such as Aesthetic, HPSv3, and diversity.

Table 1: Results of GARDO and diverse baselines across both proxy rewards and diverse o.o.d. rewards. The proxy task is marked by the orange color. ↓ and ↑ indicate improvements and decreases relative to the π_{ref}, respectively.

Method	#Step	Trained Tasks	Unseen Tasks							
		GenEval	OCR	Aesthetic	PickScore	ImgRwd	ClipScore	HPSv3	Diversity	Average
SD3.5-M [46]	-	0.63	0.58	5.07	22.40	0.83	28.2	9.70	21.84	12.66
+GRPO (β=0)	2000	0.95	0.60	4.80	21.92	0.73	28.4	6.73	15.6	11.25↓ 10.65%
+GRPO (β=0) w/o std norm	2000	0.94	0.61	4.91	22.06	0.79	28.5	6.91	15.91	11.38↓ 8.45%
+GRPO-Guard [60]	2000	0.95	0.64	5.03	22.13	0.85	28.6	7.76	17.31	11.76↓ 4.08%
+GRPO (β=0.01) w/o std norm	2000	0.81	0.64	**5.15**	**22.5**	**0.97**	28.7	**10.17**	21.73	12.84↑ 5.05%
+ GARDO (Ours) (β=0.04) w/o div	2000	0.95	0.63	5.01	22.1	0.90	28.6	8.61	19.98	12.26↓ 0.54%
+GARDO (Ours) (β=0.04)	2000	**0.95**	**0.68**	5.09	22.34	0.95	**29.4**	9.27	**24.95**	**13.24↑ 6.56%**



(a) GenEval Task (b) OCR Task

Figure 4: Learning curves and o.o.d. generalization results across different methods. GARDO not only matches the sample efficiency of the KL-free baseline, but also mitigates reward hacking effectively, as evidenced by the superior performance on unseen metrics.

Adding a KL penalty mitigates this over-optimization but at a significant cost to sample efficiency, resulting in a proxy reward that is over 10 points lower given the same computational budget. In contrast, our proposed method successfully reconciles this trade-off. As illustrated in Fig. 4, it achieves a proxy reward comparable to the strongest KL-free baseline while simultaneously preserving high unseen rewards. Notably, GARDO's generalization performance not only matches but in some cases surpasses that of the original reference model. Qualitative results on GenEval tasks are provided in Fig. 9. After training over the same steps, we find that only GARDO successfully follows the instruction and generates high-quality images, while vanilla GRPO obviously hacks the proxy reward, generating noisy images with blurred backgrounds and Gibbs artifacts.



Figure 5: Our diversity-aware advantage shaping effectively improves the generation diversity.

Diversity-Aware Advantage Shaping Improves Diversity. The results in Table 1 demonstrate that our proposed diversity-aware optimization leads to a remarkable increase in sample diversity scores, i.e., 19.98 → 24.95 for GenEval. Visualization provided in Fig. 5 further validates this efficacy. This increased diversity broadens the policy's exploration space, enabling it to discover novel states beyond the initial data distribution, and preventing convergence to a narrow set of solutions (i.e., mode collapse). Ultimately, the enhanced exploration translates into improved final performance on both proxy and unseen metrics.

Gated and Adaptive KL Enhances Sample Efficiency without Reward Hacking. As shown in Table 1 (GenEval) and Table 3 (OCR), 'GARDO w/o div', which comprises only the gated and adaptive KL regularization, is sufficient to overcome the sample efficiency bottleneck of standard regularization, matching the convergence speed of the KL-free baseline while still mitigating reward hacking. It achieves a 0.91 OCR score given 600 steps without compromising the unseen rewards (Table 3). This success is accomplished through two key components: (1) The adaptive regularization dynamically updates the reference model to prevent excessive regularization from a sub-optimal anchor, and (2) the gated KL identifies and applies a relatively high regularization specifically to "illusory" samples with high reward uncertainty, thus avoiding

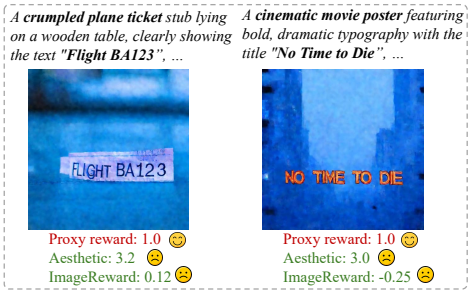
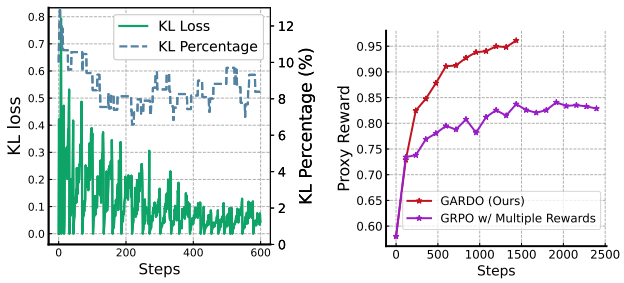


Figure 6: Samples that are identified with high uncertainty during training. While these samples reach high proxy rewards, they get low unseen rewards.

unnecessary penalties. The dynamics of KL percentage and KL loss are provided in Fig. 7a, where only around 10% of the samples are penalized. Fig. 6 shows the examples that are identified as highly uncertain and thus penalized, where the proxy reward is text rendering accuracy.

Comparison with Multiple Reward Training. While we leverage an ensemble of off-the-shelf reward models for uncertainty estimation, the policy itself is optimized against only a single proxy reward. This differentiates our approach from multi-objective reinforcement learning (MORL) methods, which seek to balance a weighted combination of multiple, often competing, reward signals. To demonstrate the superiority of our approach, we compare GARDO with the policy trained by multiple rewards, i.e., $0.8 \times \mathrm{OCR}(\mathrm{Proxy}) + 0.1 \times \mathrm{Aesthetic} + 0.1 \times \mathrm{Imagereward}$. This baseline is unregularized to maximize the sample efficiency. From the results shown in Fig. 7b, we observe that RL trained with multiple rewards exhibits significantly lower sample efficiency with respect to the primary OCR proxy reward. Moreover, multi-reward training suffers from high variance due to gradient conflicts, impeding stability and scaling up. This finding is aligned with previous multi-objective RL literature [74, 8], highlighting the challenges of optimizing for conflicting or misaligned rewards. By the way, GARDO can also optimize multiple proxy rewards simultaneously, since its core method is orthogonal to the multi-reward setup.



(a) KL dynamics (b) Compare w/ multi-reward

Figure 7: (a): Dynamics of both KL loss and gated KL percentage throughout training. (b): Reward curves of GARDO and multi-reward GRPO on the OCR task.

5.2 Emergent Behavior

By eliminating universal KL regularization for all samples and periodically resetting the reference model according to the learning dynamic, along with the diversity-aware optimization, we can effectively unlock the emergent behavior that is missing from the base model [14]. We validate this using a challenging counting task: the model is trained on datasets containing 1-9 objects and then tested on its ability to generate 10-11 objects, where the base model consistently fails. The results in Table 2 show that GARDO significantly improves the counting accuracy. This is particularly evident in the difficult task of generating 10 objects, where the base model exhibits near-zero accuracy. Fig 8 shows the corresponding qualitative examples.



Figure 8: Visualization of GARDO counting 11 objects.

5.3 Ablation Studies

Ablation over Base Gating KL Percentage k. The base KL percentage k controls the proportion of samples in each update that are selected to be regularized. Intuitively, a smaller k concentrates the constraint on the most *untrustworthy* samples (i.e., high uncertainty), whereas a larger k approaches universal regularization and risks over-constraint. As shown in Fig. 10a of Appendix B.1, smaller k consistently yield higher proxy reward and faster convergence. Increasing k to 0.5 or 1.0 substantially degrades proxy reward, indicating that applying KL too broadly can hinder optimization efficiency and over-regularize the policy.

Table 2: Counting accuracy of vanilla GRPO with different KL coefficients and our method.

Method	# Steps	Trained Tasks	Unseen Tasks	
			Counting 10	Counting 11
SD3.5-M [46]	-	0.28	0.01	0.01
GRPO ($\beta = 0.04$)	2000	0.41	0.09	0.07
GRPO ($\beta = 0.01$)	2000	0.56	0.27	0.15
GRPO ($\beta = 0$)	2000	0.77	0.28	0.15
GARDO (Ours)	2000	**0.77**	**0.38**	**0.18**

Ablation over Ensemble size K. We next study the ensemble size K used to form the auxiliary-reward consensus that drives uncertainty calculation. The employed auxiliary reward models share significantly different preferences and training data. It is highly unlikely for a generated image to hack both rewards simultaneously while remaining low variance. Larger K further provides a more reliable consensus estimate and reduces noise or bias from a single auxiliary model, but it increases the computation burden. Fig. 10b of Appendix B.1 shows that increasing K stabilizes the KL percentage k throughout training. With $K = 1$, the KL percentage k rises sharply, indicating frequent KL regularization due to noisy uncertainty estimation. Balancing performance and computational cost, we default to $K = 2$ throughout our experiments.

6 Conclusion and Limitations

In this paper, we propose GARDO, a novel and effective approach to address the challenge of reward hacking in fine-tuning diffusion models. We introduce a gated and adaptive regularization mechanism for more fine-grained control, and a diversity-aware strategy to encourage mode coverage, which significantly enhances sample efficiency and emerging behaviors. GARDO is the first formulation shifting from static and universal regularization (standard RL) to identifying trust regions tailored for visual generation tasks.

Limitations. While GARDO effectively suppresses reward hacking across all our experiments, a primary limitation is its dependency on auxiliary reward models for uncertainty estimation. Fully ruling it out is an interesting open challenge for any method that optimizes against learned reward proxies: both the primary and auxiliary reward models may be imperfect proxies for the desired behavior, and the mode-seeking nature of the RL objective can concentrate optimization on outputs that exploit these imperfections. Additionally, the scalability of our approach to resource-intensive video generative models remains an open question for future investigation. More discussion about limitations and future work is provided in Appendix C.

Acknowledgments

This work is supported by KlingAI.

References

[1] Gaon An, Seungyong Moon, Jang-Hyun Kim, and Hyun Oh Song. Uncertainty-based offline reinforcement learning with diversified q-ensemble. *Advances in neural information processing systems*, 34:7436–7447, 2021.

[2] Chenjia Bai, Lingxiao Wang, Zhuoran Yang, Zhi-Hong Deng, Animesh Garg, Peng Liu, and Zhaoran Wang. Pessimistic bootstrapping for uncertainty-driven offline reinforcement learning. In *International Conference on Learning Representations*, 2022.

[3] Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *ICLR*, 2024.

[4] Jingye Chen, Yupan Huang, Tengchao Lv, Lei Cui, Qifeng Chen, and Furu Wei. Textdiffuser: Diffusion models as text painters. *Advances in Neural Information Processing Systems*, 36:9353–9387, 2023.

[5] Kevin Clark, Paul Vicol, Kevin Swersky, and David J. Fleet. Directly fine-tuning diffusion models on differentiable rewards. In *The Twelfth International Conference on Learning Representations*, 2024.

[6] William R Clements, Bastien Van Delft, Benoît-Marie Robaglia, Reda Bahi Slaoui, and Sébastien Toth. Estimating risk and uncertainty in deep reinforcement learning. *arXiv preprint arXiv:1905.09638*, 2019.

[7] Haoyou Deng, Keyu Yan, Chaojie Mao, Xiang Wang, Yu Liu, Changxin Gao, and Nong Sang. Densegrpo: From sparse to dense reward for flow matching model alignment. *arXiv preprint arXiv:2601.20218*, 2026.

[8] Jacob Eisenstein, Chirag Nagpal, Alekh Agarwal, Ahmad Beirami, Alex D'Amour, DJ Dvijotham, Adam Fisch, Katherine Heller, Stephen Pfohl, Deepak Ramachandran, et al. Helping or herding? reward model ensembles mitigate but do not eliminate reward hacking. *arXiv preprint arXiv:2312.09244*, 2023.

[9] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

[10] Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 36:79858–79885, 2023.

[11] Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Reinforcement learning for fine-tuning text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

[12] Xiaolong Fu, Lichen Ma, Zipeng Guo, Gaojing Zhou, Chongxiao Wang, ShiPing Dong, Shizhe Zhou, Ximan Liu, Jingling Fu, Tan Lit Sin, et al. Dynamic-treerpo: Breaking the independent trajectory bottleneck with structured sampling. *arXiv preprint arXiv:2509.23352*, 2025.

[13] Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36:52132–52152, 2023.

[14] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Nature*, 645(8081):633–638, Sep 2025.

[15] Shashank Gupta, Chaitanya Ahuja, Tsung-Yu Lin, Sreya Dutta Roy, Harrie Oosterhuis, Maarten de Rijke, and Satya Narayan Shukla. A simple and effective reinforcement learning method for text-to-image diffusion fine-tuning. *arXiv preprint arXiv:2503.00897*, 2025.

[16] Anthony GX-Chen, Jatin Prakash, Jeff Guo, Rob Fergus, and Rajesh Ranganath. Kl-regularized reinforcement learning is designed to mode collapse. *arXiv preprint arXiv:2510.20817*, 2025.

[17] Haoran He, Jiajun Liang, Xintao Wang, Pengfei Wan, Di Zhang, Kun Gai, and Ling Pan. Scaling image and video generation via test-time evolutionary search. *arXiv preprint arXiv:2505.17618*, 2025.

[18] Xiaoxuan He, Siming Fu, Yuke Zhao, Wanli Li, Jian Yang, Dacheng Yin, Fengyun Rao, and Bo Zhang. Tempflow-grpo: When timing matters for grpo in flow models. *arXiv preprint arXiv:2508.04324*, 2025.

[19] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*, 2021.

[20] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

[21] Sujie Hu, Chubin Chen, Jiashu Zhu, Jiahong Wu, Xiangxiang Chu, and Xiu Li. Embedding-perturbed exploration preference optimization for flow models. In *Forty-third International Conference on Machine Learning*, 2026.

[22] Xuepeng Jing, Wenhuan Lu, Hao Meng, Zhizhi Yu, and Jianguo Wei. Tigflow-grpo: Trajectory forecasting via interaction-aware flow matching and reward-driven optimization. *arXiv preprint arXiv:2603.24936*, 2026.

[23] Jaihoon Kim, Taehoon Yoon, Jisung Hwang, and Minhyuk Sung. Inference-time scaling for flow models via stochastic generation and rollover budget forcing. *arXiv preprint arXiv:2503.19385*, 2025.

[24] Sunwoo Kim, Minkyu Kim, and Dongmin Park. Test-time alignment of diffusion models without reward over-optimization. In *The Thirteenth International Conference on Learning Representations*, 2025.

[25] Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *Advances in neural information processing systems*, 36:36652–36663, 2023.

[26] Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024.

[27] Cassidy Laidlaw, Shivam Singhal, and Anca Dragan. Correlated proxies: A new definition and improved mitigation for reward hacking. In *The Thirteenth International Conference on Learning Representations*, 2025.

[28] Jeongjae Lee, Jinho Chang, Jeongsol Kim, and Jong Chul Ye. Reward score matching: Unifying reward-based fine-tuning for flow and diffusion models. *arXiv preprint arXiv:2604.17415*, 2026.

[29] Junzhe Li, Yutao Cui, Tao Huang, Yinping Ma, Chun Fan, Miles Yang, and Zhao Zhong. Mixgrpo: Unlocking flow-based grpo efficiency with mixed ode-sde. *arXiv preprint arXiv:2507.21802*, 2025.

[30] Xiner Li, Yulai Zhao, Chenyu Wang, Gabriele Scalia, Gokcen Eraslan, Surag Nair, Tommaso Biancalani, Shuiwang Ji, Aviv Regev, Sergey Levine, et al. Derivative-free guidance in continuous and discrete diffusion models with soft value-based decoding. *arXiv preprint arXiv:2408.08252*, 2024.

[31] Zhanhao Liang, Tao Yang, Jie Wu, Chengjian Feng, and Liang Zheng. Leapalign: Post-training flow matching models at any generation step by building two-step trajectories. *arXiv preprint arXiv:2604.15311*, 2026.

[32] Zhanhao Liang, Yuhui Yuan, Shuyang Gu, Bohan Chen, Tiankai Hang, Ji Li, and Liang Zheng. Step-aware preference optimization: Aligning preference with denoising performance at each step. *arXiv preprint arXiv:2406.04314*, 2024.

[33] Henglin Liu, Huijuan Huang, Jing Wang, Chang Liu, Xiu Li, and Xiangyang Ji. Diverseg-rpo: Mitigating mode collapse in image generation via diversity-aware grpo. *arXiv preprint arXiv:2512.21514*, 2025.

[34] Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di ZHANG, and Wanli Ouyang. Flow-GRPO: Training flow matching models via online RL. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[35] Jie Liu, Gongye Liu, Jiajun Liang, Ziyang Yuan, Xiaokun Liu, Mingwu Zheng, Xiele Wu, Qiulin Wang, Menghan Xia, Xintao Wang, et al. Improving video generation with human feedback. *arXiv preprint arXiv:2501.13918*, 2025.

[36] Yihong Luo, Tianyang Hu, and Jing Tang. Reinforcing diffusion models by direct group preference optimization. *arXiv preprint arXiv:2510.08425*, 2025.

[37] Yuhang Ma, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. Hpsv3: Towards wide-spectrum human preference score. *International Conference on Computer Vision*, 2025.

[38] Zichen Miao, Jiang Wang, Ze Wang, Zhengyuan Yang, Lijuan Wang, Qiang Qiu, and Zicheng Liu. Training diffusion models towards diverse image generation with reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10844–10853, 2024.

[39] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. *Advances in neural information processing systems*, 29, 2016.

[40] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[41] Alexander Pan, Kush Bhatia, and Jacob Steinhardt. The effects of reward misspecification: Mapping and mitigating misaligned models. In *International Conference on Learning Representations*, 2022.

[42] Bowen Ping, Chengyou Jia, Minnan Luo, Hangwei Qian, and Ivor Tsang. Flow-factory: A unified framework for easy reinforcement learning in flow-matching models.

[43] Mihir Prabhudesai, Anirudh Goyal, Deepak Pathak, and Katerina Fragkiadaki. Aligning text-to-image diffusion models with reward backpropagation. *arXiv preprint arXiv:2310.03739*, 2023.

[44] You Qin, Linqing Wang, Hao Fei, Roger Zimmermann, Liefeng Bo, Qinglin Lu, and Chunyu Wang. Soar: Self-correction for optimal alignment and refinement in diffusion models. *arXiv preprint arXiv:2604.12617*, 2026.

[45] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023.

[46] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[47] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

[48] Yash Savani, Branislav Kveton, Yuchen Liu, Yilin Wang, Jing Shi, Subhojyoti Mukherjee, Nikos Vlassis, and Krishna Kumar Singh. Stepwise credit assignment for grpo on flow-matching models. *arXiv preprint arXiv:2603.28718*, 2026.

[49] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294, 2022.

[50] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[51] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[52] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

[53] Saurabh Singh and Ian Fischer. Stochastic sampling from deterministic flow models. *arXiv preprint arXiv:2410.02217*, 2024.

[54] Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward gaming. *Advances in Neural Information Processing Systems*, 35:9460–9471, 2022.

[55] Bingda Tang, Yuhui Zhang, Xiaohan Wang, Jiayuan Mao, Ludwig Schmidt, and Serena Yeung-Levy. V-grpo: Online reinforcement learning for denoising generative models is easier than you think. *arXiv preprint arXiv:2604.23380*, 2026.

[56] Masatoshi Uehara, Yulai Zhao, Tommaso Biancalani, and Sergey Levine. Understanding reinforcement learning-based fine-tuning of diffusion models: A tutorial and review. *arXiv preprint arXiv:2407.13734*, 2024.

[57] Masatoshi Uehara, Yulai Zhao, Kevin Black, Ehsan Hajiramezanali, Gabriele Scalia, Nathaniel Lee Diamant, Alex M Tseng, Tommaso Biancalani, and Sergey Levine. Fine-tuning of continuous-time diffusion models as entropy-regularized control. *arXiv preprint arXiv:2402.15194*, 2024.

[58] Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8228–8238, 2024.

[59] Feng Wang and Zihao Yu. Coefficients-preserving sampling for reinforcement learning with flow matching. *arXiv preprint arXiv:2509.05952*, 2025.

[60] Jing Wang, Jiajun Liang, Jie Liu, Henglin Liu, Gongye Liu, Jun Zheng, Wanyuan Pang, Ao Ma, Zhenyu Xie, Xintao Wang, et al. Grpo-guard: Mitigating implicit over-optimization in flow matching via regulated clipping. *arXiv preprint arXiv:2510.22319*, 2025.

[61] Yibin Wang, Zhimin Li, Yuhang Zang, Yujie Zhou, Jiazi Bu, Chunyu Wang, Qinglin Lu, Cheng Jin, and Jiaqi Wang. Pref-grpo: Pairwise preference reward-based grpo for stable text-to-image reinforcement learning. *arXiv preprint arXiv:2508.20751*, 2025.

[62] Yibin Wang, Yuhang Zang, Hao Li, Cheng Jin, and Jiaqi Wang. Unified reward model for multimodal understanding and generation. *arXiv preprint arXiv:2503.05236*, 2025.

[63] Yibin Wang, Yuhang Zang, Hao Li, Cheng Jin, and Jiaqi Wang. Unified reward model for multimodal understanding and generation. *arXiv preprint arXiv:2503.05236*, 2025.

[64] Olivia Wiles, Sven Gowal, Florian Stimberg, Sylvestre Alvise-Rebuffi, Ira Ktena, Krishnamurthy Dvijotham, and Taylan Cemgil. A fine-grained analysis on distribution shift. *arXiv preprint arXiv:2110.11328*, 2021.

[65] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

[66] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report, 2025.

[67] Jie Wu, Yu Gao, Zilyu Ye, Ming Li, Liang Li, Hanzhong Guo, Jie Liu, Zeyue Xue, Xiaoxia Hou, Wei Liu, et al. Rewarddance: Reward scaling in visual generation. *arXiv preprint arXiv:2509.08826*, 2025.

[68] Luhuan Wu, Brian Trippe, Christian Naesseth, David Blei, and John P Cunningham. Practical and asymptotically exact conditional sampling in diffusion models. *Advances in Neural Information Processing Systems*, 36:31372–31403, 2023.

[69] Enze Xie, Junsong Chen, Yuyang Zhao, Jincheng Yu, Ligeng Zhu, Chengyue Wu, Yujun Lin, Zhekai Zhang, Muyang Li, Junyu Chen, et al. Sana 1.5: Efficient scaling of training-time and inference-time compute in linear diffusion transformer. *arXiv preprint arXiv:2501.18427*, 2025.

[70] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023.

[71] Shuchen Xue, Chongjian Ge, Shilong Zhang, Yichen Li, and Zhi-Ming Ma. Advantage weighted matching: Aligning rl with pretraining in diffusion models. *arXiv preprint arXiv:2509.25050*, 2025.

[72] Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, et al. Dancegrpo: Unleashing grpo on visual generation. *arXiv preprint arXiv:2505.07818*, 2025.

[73] Kai Yang, Jian Tao, Jiafei Lyu, Chunjiang Ge, Jiaxin Chen, Weihan Shen, Xiaolong Zhu, and Xiu Li. Using human feedback to fine-tune diffusion models without any reward model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8941–8951, 2024.

[74] Runzhe Yang, Xingyuan Sun, and Karthik Narasimhan. A generalized algorithm for multi-objective reinforcement learning and policy adaptation. *Advances in neural information processing systems*, 32, 2019.

[75] Benjamin Yu, Jackie Liu, and Justin Cui. Smart-grpo: Smartly sampling noise for efficient rl of flow-matching models. *arXiv preprint arXiv:2510.02654*, 2025.

[76] Huizhuo Yuan, Zixiang Chen, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning of diffusion models for text-to-image generation. *arXiv preprint arXiv:2402.10210*, 2024.

[77] Jiacheng Zhang, Jie Wu, Weifeng Chen, Yatai Ji, Xuefeng Xiao, Weilin Huang, and Kai Han. Onlinevpo: Align video diffusion model with online video-centric preference optimization. *arXiv preprint arXiv:2412.15159*, 2024.

[78] Liyu Zhang, Kehan Li, Tingrui Han, Tao Zhao, Yuxuan Sheng, Shibo He, and Chao Li. Op-grpo: Efficient off-policy grpo for flow-matching models. *arXiv preprint arXiv:2604.04142*, 2026.

[79] Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, Kai Jiang, Hang Su, Stefano Ermon, Jun Zhu, and Ming-Yu Liu. Diffusionnft: Online diffusion reinforcement with forward process. *arXiv preprint arXiv:2509.16117*, 2025.

[80] Yujie Zhou, Pengyang Ling, Jiazi Bu, Yibin Wang, Yuhang Zang, Jiaqi Wang, Li Niu, and Guangtao Zhai. G2rpo: Granular grpo for precise reward in flow models. *arXiv preprint arXiv:2510.01982*, 2025.

A Implementation Details

A.1 Details of GARDO

We choose Flow-GRPO [34] as the base RL algorithm to implement GARDO. We provide the training details and related hyperparameters as follows:

- Following Flow-GRPO [34], we set the sampling timestep $T = 10$ and an evaluation timestep $T = 40$.
- We set the training batch size as 6, and set the group size $G = 24$.
- Learning rate is $3 \cdot 10^{-4}$, and the clip range is $1e^{-4}$.
- We use Lora with $\alpha = 32$ and $r = 64$.
- Given a group of images generated with the same prompt, the initial noises are set to be the same to ensure GRPO's assumptions.

GARDO's adaptive KL. We set the KL threshold $\epsilon_{\text{KL}} = 1e - 4$. The maximum gradient steps m between two consecutive reset operations is set to be 100. This means reset happens if the KL loss surpasses ϵ_{KL} or there have been m gradient steps after the last recent reset.

GARDO's gated KL. We set the initial KL percentage $k = 0.1$. This means the top 10% samples with the highest uncertainty $\mathcal{U}$ are penalized at the beginning of the training stage. k is dynamically updated throughout the training process. We maintain a cache window to determine k. The window size is 20, which stores the maximum and minimum value of uncertainty $\mathcal{U}$ over 20 epochs. If the mean value of the uncertainty within the current batch is higher than the maximum value cached in the window, then we set $k \leftarrow k \times 1.1$; if the mean value of the uncertainty within the current batch is lower than the minimum value cached in the window, $k \leftarrow k \times 0.9$; otherwise, k remains unchanged. We summarize our method in the pseudocode in Alg. 1.

A.2 Computation Specification

We train our model using 8 NVIDIA A800 GPUs. It usually takes around 49s for a gradient step, including both sampling and training.

A.3 Evaluation Metrics

Proxy Tasks. In this paper, we include two proxy tasks to train the diffusion models by RL, i.e., GenEval and OCR tasks. We ensure the training hyperparameters are the same across different methods for fair comparison. Below, we introduce the proxy tasks used in the paper.

- The GenEval framework follows Flow-GRPO's experimental protocol. The training dataset is sourced from the Flow-GRPO dataset. GenEval includes six difficult compositional image generation tasks. We use its official evaluation pipeline, which detects object co-occurrence, spatial positioning, object count, and color attributes for fine-grained assessment.
- For the OCR task, we use the training dataset and test dataset from Flow-GRPO. This task measures text rendering accuracy with the reward $r = \max(1 - Ne/N_{ref}, 0)$, where Ne is the minimum edit distance between the rendered text and the target text and N_{ref} is the number of characters required to render.

Unseen Tasks. We adopt DrawBench [47] for evaluation, which consists of 200 prompts spanning 11 different categories, serving as an effective test set for comprehensive evaluation of the T2I models. For each prompt, we generate four images for evaluation for a fair and convincing comparison. We employ Aesthetic [49], PickScore [25], ImageReward [70], ClipScore [19], and HPSv3 [37] for extensive evaluation of the o.o.d. generalization ability, and detecting the degree of reward hacking. For the diversity score, we employ Dinov3 [52] to extract feature embeddings e^i of each image. Then we use the mean of pairwise cosine distance across a group of images for diversity quantification:

$$\text{Div} = \text{mean}_{i,j\in[1,G],i\neq j}(1 - \frac{e_i \cdot e_j}{|e_i||e_j|}), \tag{9}$$

where a group of images is generated, given the same prompt.

Algorithm 1 Overview of GARDO

Require: initial policy model π_θ; proxy reward $\tilde{R}$; auxiliary reward models $\hat{R}_1, \hat{R}_2$; Dinov3 model f_ϕ; KL loss threshold ϵ_{KL}; maximum reset steps m; prompt dataset $\mathcal{C}$; total sampling steps $\tilde{T}$; number of samples per prompt G; Gated KL percentage k.

Set reference policy $\pi_{\text{ref}} \leftarrow \pi_\theta$
Uncertainty window $\mathcal{W}_\mathcal{U} \leftarrow \emptyset$ with a window size W
for training iteration $n = 1$ **to** N **do**
  Sample batch prompts $\mathcal{C}_b \sim \mathcal{C}$
  Update old policy model: $\pi_{\theta_{\text{old}}} \leftarrow \pi_\theta$
  $\mathcal{L}_{total} \leftarrow 0, \bar{\mathcal{U}}_{batch} \leftarrow 0$
  for each prompt $\mathbf{c} \in \mathcal{C}_b$ **do**
    Initial the same noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$
    Generate G images $\{x_0\}_{i=1}^G$ after $\tilde{T}$ sampling steps.
    Obtains advantages $A_{i=1}^G$ from $\tilde{R}$ via Eq. 1.
    Reshape $A_{i=1}^G$ and get diversity-aware advantages by Eq. 8.
    Obtains uncertainty estimation $\mathcal{U}_{i=1}^G$ via Eq. 7.
    Determine gate threshold: $\epsilon_\mathcal{U} = \text{Percentile}(\{\mathcal{U}^i\}_{i=1}^G, 1-k)$.
    Calculate KL-regularized loss $\mathcal{L}_{total} \leftarrow \mathcal{L}_{total} + \frac{1}{G}\sum_{i=1}^G [\mathcal{L}_{\text{RL}}^i + \mathbb{I}(\mathcal{U}^i > \epsilon_\mathcal{U}) \cdot \mathcal{L}_{\text{KL}}^i]$
    $\bar{\mathcal{U}}_{batch} \leftarrow \bar{\mathcal{U}}_{batch} + (\text{mean}(\{\mathcal{U}^i\}))$
  end for
  Update policy model via gradient ascent: $\theta \leftarrow \theta + \eta \nabla_\theta \mathcal{L}_{total}$
  if $n \bmod m$ is 0 or $\mathcal{L}_{\text{KL}} > \epsilon_{\text{KL}}$ **then**
    Reset $\pi_{\text{ref}} \leftarrow \pi_\theta$
  end if
  if $|\mathcal{W}_\mathcal{U}| \geq W$ **then**
    if $\bar{\mathcal{U}}_{batch} > \max(\mathcal{W}_\mathcal{U})$ **then**
      $k \leftarrow \min(k \times 1.1, 1.0)$
    end if
    if $\bar{\mathcal{U}}_{batch} < \min(\mathcal{W}_\mathcal{U})$ **then**
      $k \leftarrow k \times 0.9$
    end if
  end if
  Update window $\mathcal{W}_\mathcal{U}$: Append $\bar{\mathcal{U}}_{batch}$, remove oldest if $|\mathcal{W}_\mathcal{U}| > W$.
end for

B Additional Results

B.1 Additional Main Results

Table 3: Results of GARDO and diverse baselines across both proxy rewards and diverse o.o.d. rewards. The proxy task is marked by the orange color.

Method	#Step	Trained Tasks	Unseen Tasks							
		OCR	GenEval	Aesthetic	PickScore	ImgRwd	ClipScore	HPSv3	Diversity	Average
SD3.5-M [46]	-	0.58	0.63	5.07	22.40	0.83	28.2	9.70	21.84	12.67
+GRPO (β=0)	600	<u>0.93</u>	0.52	4.67	21.82	0.61	27.9	8.11	18.15	11.68
+GRPO (β=0) w/o std norm	600	0.92	0.57	4.88	22.05	0.68	28.0	8.32	19.37	11.98
+GRPO (β=0.01) w/o std norm	600	0.86	<u>0.64</u>	**5.08**	**22.45**	<u>0.90</u>	28.6	**9.89**	<u>21.32</u>	<u>12.70</u>
+ GARDO (Ours) (β=0.04) w/o div	600	0.91	0.63	5.03	22.41	0.87	28.7	9.22	19.89	12.39
+GARDO (Ours) (β=0.04)	600	0.92	**0.65**	**5.07**	**22.41**	**0.92**	**28.7**	<u>9.75</u>	**21.60**	**12.73**
+GARDO (Ours) (β=0.04)	1400	**0.96**	0.60	5.02	22.31	0.87	**28.7**	9.51	21.03	12.58

B.2 Results on DiffusionNFT [79]

GARDO is built upon an existing regularized-RL objective, which can be compatible with various RL algorithms. To demonstrate GARDO's versatility, we apply it to a recently released RL algorithm, DiffusionNFT [79], which can be very different from GRPO. DiffusionNFT directly optimizes velocity without relying on the computation of log likelihood, bridging the gap between SFT pre-training and RL post-training. From the results shown in Table 4, we have the following observations: (1) Similar to Flow-GRPO, vanilla DiffusionNFT still suffers from reward hacking, as evidenced

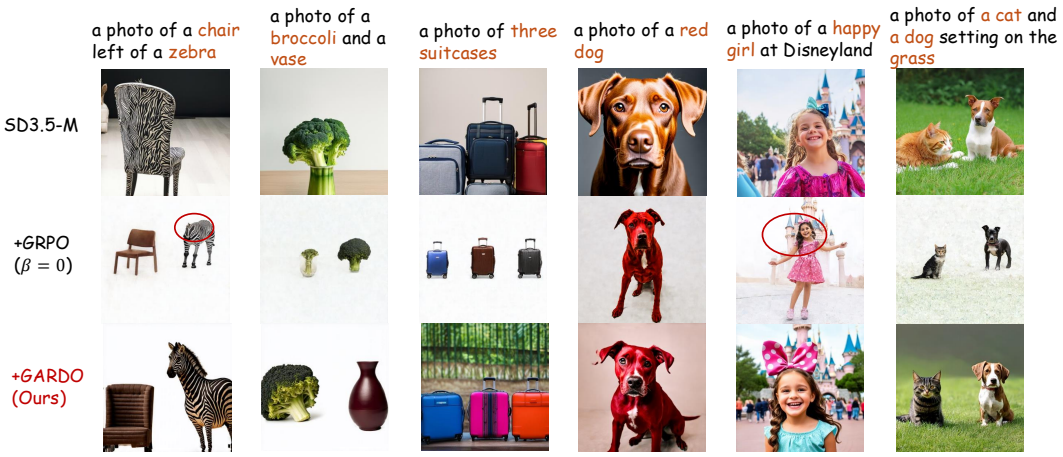
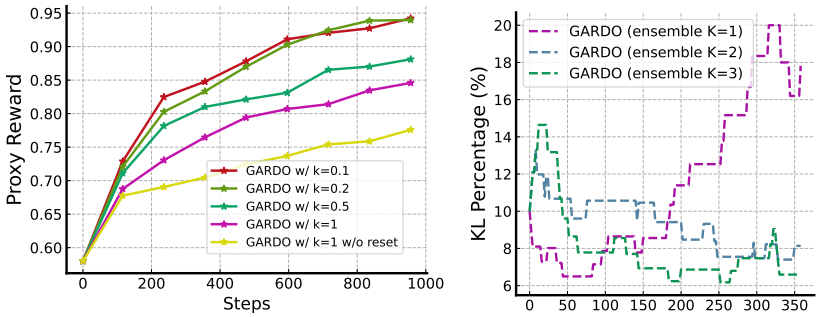


Figure 9: Qualitative images generated by GARDO and vanilla GRPO across different prompts. Only GARDO generates images correctly aligned with the prompt, while maintaining a satisfactory perceptual quality and diversity.



(a) Gating fraction k (b) Ensemble size K

Figure 10: (a): Performance comparison of GARDO variants on the OCR task. k indicates the base gating KL percentage. GARDO without periodically resetting π_{ref} suffers from reduced sample efficiency. (b) KL dynamics over the training process. An insufficient ensemble size K leads to an increase in uncertain samples and instability.

by the reduced performance on unseen tasks like Aesthetic, PickScore, and HPSv3. (2) Adding KL regularization can effectively mitigate reward hacking, enabling more robust optimization. However, it significantly compromises the sample efficiency. Given the same training steps (i.e., 400 steps), KL-regularized DiffusionNFT only achieves a 0.72 accuracy on the proxy task, i.e., GenEval. (3) GARDO performs best in a balance between sample efficiency and the mitigation of reward hacking. GARDO can obtain the highest score (i.e., 0.95) on GenEval given 400 steps without reducing unseen rewards compared to the reference model. It even boosts the unseen reward, like Aesthetic, and remains the highest diversity compared with baselines. This clearly demonstrates that GARDO achieves the highest sample efficiency while effectively preventing hacking, showing superior performance in terms of both proxy rewards and unseen rewards.

B.3 Comparison with Pref-GRPO [61]

We compare with Pref-GRPO [61] following its official implementation and configuration. We use the same training dataset and test dataset as used in Pref-GRPO. Flux.1-dev is employed as the base model. For both methods, we evaluate the checkpoints after 300 training steps to ensure fair and fast comparison. Results demonstrate that GARDO attains superior performance in both i.i.d. rewards (UniRew) and OOD metrics, achieving **faster** training time.

Table 4: Results on GenEval tasks for DiffusionNFT algorithm. GARDO successfully surpasses baselines in terms of both sample efficiency in proxy reward and generalization on o.o.d. rewards.

Method	#Step	Trained Tasks		Unseen Tasks					
		GenEval	OCR	Aesthetic	PickScore	ImgRwd	ClipScore	HPSv3	Diversity
SD3.5-M [46]	-	0.63	0.58	5.07	22.40	0.83	28.2	9.70	21.84
+DiffusionNFT (β=0)	400	0.94	0.68	4.23	21.85	0.62	28.8	5.66	11.78
+DiffusionNFT (β=0.04)	400	0.72	0.54	4.87	22.54	1.14	29.1	11.51	13.87
+DiffusionNFT (β=0.04)	1200	0.90	0.51	4.55	21.96	0.89	29.0	10.04	13.70
+GARDO (Ours) (β=0.04)	400	0.95	**0.64**	**5.23**	**22.59**	**1.10**	**29.2**	12.17	**14.57**

Table 5: Following the official setup of Pref-GRPO, GARDO outperforms Pref-GRPO with less training time in terms of both i.i.d. and o.o.d. metrics.

Method	Step	Time	UniRew	ImgRew	PickScore	Aesthetic
Pref-GRPO	300	27.4h	3.05	1.303	22.69	6.163
GARDO (ours)	300	**19.1h**	**3.11**(↑)	**1.348**(↑)	**22.80**(↑)	**6.223**(↑)

B.4 Results on Flux.1-dev

GARDO can also generalize to different base models. We choose Flux.1-dev [26] as the base model, which contains 12B parameters and is known as a strong model for text-to-image generation. Here, we leverage HPSv2 as the proxy reward for optimization, using open-sourced HPDv3 (https://huggingface.co/datasets/MizzenAI/HPDv3) as the training dataset. GARDO still achieves the highest sample efficiency compared with the KL-regularized Flow-GRPO method. We provide the reward curves in Fig. 11 and qualitative demos in Fig. 13 and Fig. 14. As shown in Fig. 12, GARDO outperforms both Flow-GRPO and base models in generalizing to unseen rewards, indicating GARDO's great potential to optimize any proxy rewards without reward hacking.

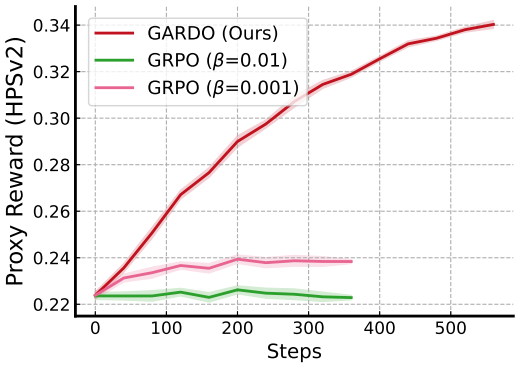


Figure 11: Learning curves of GARDO on Flux.1-dev with HPSv2 as the proxy reward.

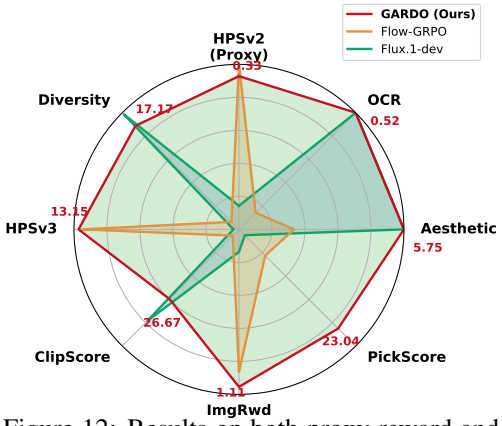


Figure 12: Results on both proxy reward and o.o.d. unseen rewards. GARDO performs best in the trade-off.

B.5 Results on Generation Diversity

Following the evaluation protocol of DiverseGRPO [33], we evaluate diversity using DreamSim, FID, and SSIM, which are not used as training rewards, and report the mean and standard deviation over three random seeds. This experiment examines whether GARDO can improve proxy-task performance while preserving diversity. As shown in Table 6, GARDO achieves higher DreamSim and DINOv3 diversity, lower FID, and higher SSIM than vanilla GRPO, with a comparable GenEval score. Removing the diversity component reduces DreamSim and DINOv3 diversity.

B.6 More Qualitative Results

We provide the generated images along the training process in Fig. 15 and Fig. 16. As the training step increases, we observe that Flow-GRPO obviously hacks the reward (or exploits the flaws), yielding reduced perceptual visual quality. However, GARDO remains a high visual quality throughout the training process, without compromising optimization performance on the proxy reward.

Table 6: Diversity and proxy-task results. Results for trained methods are reported as mean ± standard deviation over three random seeds.

Method	DreamSim ↑	FID ↓	SSIM ↑	DINOv3 Div. ↑	GenEval ↑
Base model	0.156	–	–	21.84	0.91
GRPO	0.101 ± 0.004	57.4 ± 0.3	0.151 ± 0.008	15.7 ± 0.062	0.95 ± 0.007
GARDO w/o diversity shaping	0.127 ± 0.004	49.8 ± 0.2	0.174 ± 0.010	19.5 ± 0.088	0.95 ± 0.010
GARDO (Ours)	$\mathbf{0.152 \pm 0.005}$	$\mathbf{44.3 \pm 0.2}$	$\mathbf{0.182 \pm 0.010}$	$\mathbf{24.8 \pm 0.121}$	0.95 ± 0.008



Figure 13: Qualitative results on Flux.1-dev across GARDO and baselines. While both GRPO and GARDO significantly improve the visual quality, GRPO tends to hack the HPSv2 reward, generating unnecessary or even undesired details.

C Broader Impact, Discussions, and Limitations

Broader Impact. GARDO could have a positive societal impact by making text-to-image diffusion models more reliable during reinforcement learning fine-tuning, reducing reward hacking where models optimize proxy metrics while producing low-quality, misleading, or collapsed outputs. By improving diversity and maintaining image quality, the method can support safer deployment of generative models in creative, educational, and commercial applications where alignment with human preferences matters.



Figure 14: Qualitative results on Flux.1-dev across GARDO and baselines. While both GRPO and GARDO significantly improve the visual quality, GRPO tends to hack the HPSv2 reward, generating unnecessary or even undesired details.

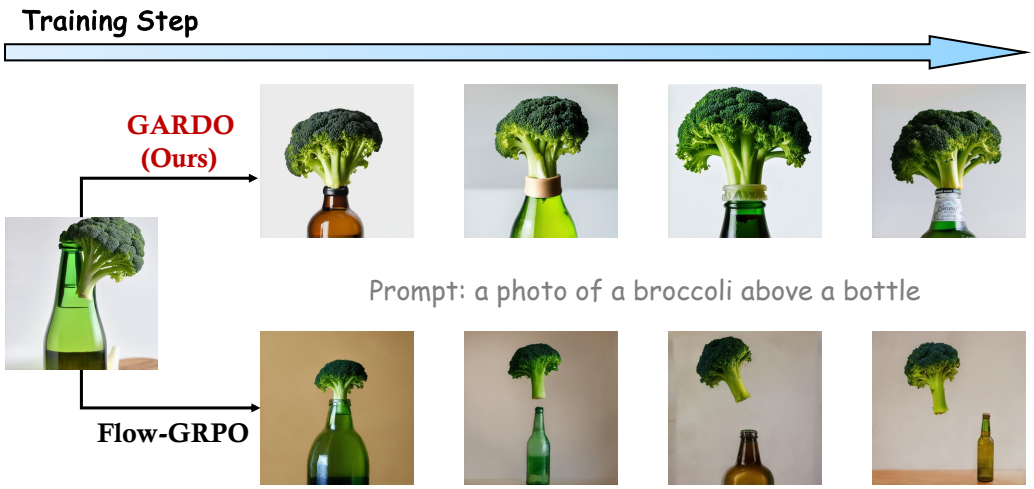


Figure 15: Generated images along the training process.

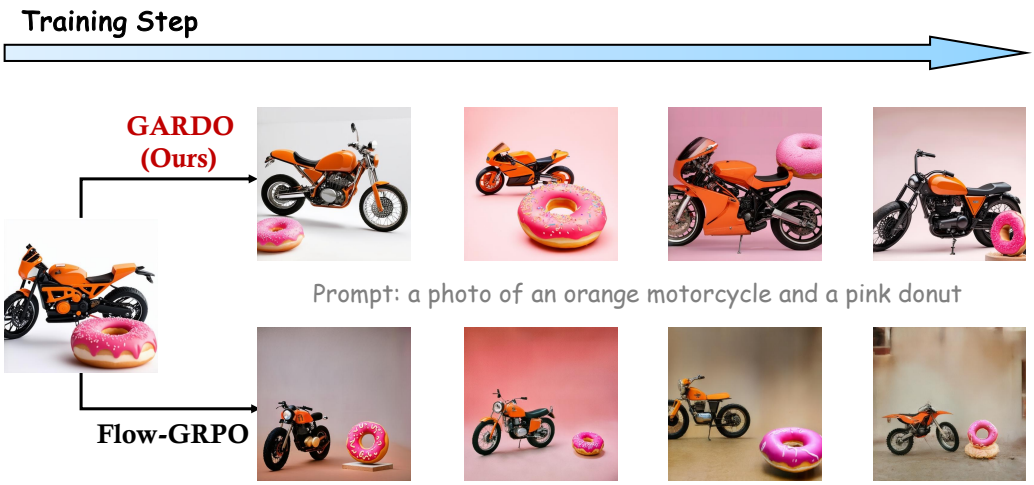


Figure 16: Generated images along the training process.

Hyperparameter Robustness. GARDO introduces several design choices, including the uncertainty-gating threshold for KL regularization and the adaptive trust region schedule. In principle, these hyperparameters could affect the trade-off between reward maximization, exploration, and regularization. In our experiments, we intentionally fix the hyperparameters across all tasks, reward settings, and model variants rather than tuning them separately for each task. This design choice is meant to evaluate the robustness and practical usability of GARDO under a realistic setting, where extensive per-task hyperparameter search is often infeasible. The consistently strong performance across proxy rewards, unseen reward models, and different base models suggests that GARDO is not sensitive to certain hyperparameter choices. Moreover, we provide an ablation study on the effects of hyperparameters in Fig. 10.

Limitations of Auxiliary Reward Models. GARDO relies on auxiliary reward models to estimate reward uncertainty and to identify samples where optimizing the target proxy reward may lead to reward hacking. This introduces an inherent dependency on the quality, coverage, and biases of the auxiliary reward models. If the auxiliary reward models are poorly calibrated, biased toward certain visual styles, or misaligned with the target domain, the uncertainty estimate may become inaccurate. In such cases, GARDO may either over-regularize genuinely good samples that are not favored by the auxiliary models, or under-regularize samples that exploit shared weaknesses across reward models. However, this limitation is also present in most practical reward-guided generation and alignment pipelines, where no single reward model perfectly represents human preference. GARDO does not require auxiliary models to be individually perfect; rather, it uses their disagreement with the optimized proxy reward as a lightweight signal for potential reward hacking. In real-world scenarios, it is usually feasible to obtain several complementary reward signals, such as aesthetic predictors, image-text alignment models, preference models, OCR or compositionality evaluators, and task-specific quality metrics. Even if each auxiliary reward model is imperfect, their combined signal is often sufficient to detect obvious distributional shifts or pathological reward exploitation. Therefore, while auxiliary model bias remains a limitation, GARDO provides a practical mechanism for leveraging available imperfect signals to improve robustness over optimizing a single proxy reward alone.

Adaptive Trust Region. The adaptive trust region improves sample efficiency in practice, but its long-term behavior under adversarial or highly biased rewards warrants further theoretical study. Future work could develop a more formal framework that relates reward model disagreement to bounds on proxy-reward error, analyzes selective regularization under imperfect uncertainty estimates, and characterizes when adaptive reference updates preserve alignment rather than amplifying reward hacking.

Evaluation. A further limitation is that our evaluation primarily relies on automatic metrics, including both the optimized proxy reward and held-out reward models. This evaluation protocol is useful because it allows scalable, repeatable, and controlled comparison across methods. In particular, evaluating on unseen reward models provides evidence that GARDO does not merely overfit the

optimized reward and can generalize to related notions of image quality and alignment. Nevertheless, automatic reward models remain imperfect proxies for human judgment. Strong performance on held-out metrics does not fully rule out subtle reward hacking, nor does it guarantee that generated images are preferred by humans in all cases. Human evaluation would provide a more direct assessment of whether GARDO improves perceived visual quality, prompt alignment, diversity, and overall preference. We view human evaluation as an important direction for future work. Despite this limitation, the consistent improvements across multiple held-out metrics suggest that GARDO is a promising step toward reducing reward hacking in diffusion model reinforcement learning.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The contributions of this paper are summarized in the introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in the conclusion section.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provide a formal definition for each proposition.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We illustrate the details of our method in Appendix A. We also provide the code to reproduce our experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide the anonymous code of our proposed method.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We summarize all the details in Appendix A and experiment section.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We follow official evaluation setup in previous works [34] to provide fair and faithful comparison.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide these details in Appendix A.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We follow the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss it in the broader impact section.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper does not release data or models.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Related works are properly cited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We provide a detailed Readme.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method of this paper does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.